

2020

# Identifying Online Sexual Predators Using Support Vector Machine

Yifan Li

*Technological University Dublin*

# Identifying Online Sexual Predators using Support Vector Machine


**Yifan Li**

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Analytics)

**2020-01-26**

# Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Stream), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

***Signed:*** Yifan Li

***Date:2010-01-26***

# Abstract

A two-stage classification model is built in the research for online sexual predator identification. The first stage identifies the suspicious conversations that have predator participants. The second stage identifies the predators in suspicious conversations. Support vector machines are used with word and character n-grams, combined with behavioural features of the authors to train the final classifier. The unbalanced dataset is downsampled to test the performance of re-balancing an unbalanced dataset. An age group classification model is also constructed to test the feasibility of extracting the age profile of the authors, which can be used as features for classifier training.

The effect of re-balancing the unbalanced dataset resulted in a better performance of the classifier. Testing the two-stage classification model on the unseen test set, 171 out of 254 predators are successfully identified giving a precision of 0.85, recall of 0.67 and f-score of 0.807. Comparing the classification performance with and without the behavioural feature, it can be seen the n-gram contributed the most to the performance of the classifier, while the behavioural features do not contribute significantly to the performance.

**Keywords:** Text classification, support vector machine, unbalanced dataset, predator identification, age classification

# Acknowledgments

I would like to thank my supervisor Dr. Séan O'Leary who helped me, and guided me through the course of the dissertation.

I would also like to thank Dr. Luca Longo, who helped me developing the initial ideas of this dissertation with his knowledge and experience.

Finally, I would like to thank all my professors and friends at TU Dublin, my life would not be the same without you.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **SVM** | Support Vector Machine |
| **CLEF** | Conference and Labs of the Evaluation Forum |
| **PJ** | Perverted Justice |
| **IRC** | Internet Relay Chat |
| **NLP** | Natural Language Processing |
| **API** | Application Programming Interface |
| **TP** | True Positive |
| **FP** | False Positive |
| **TN** | True Negative |
| **FN** | False Negative |
| **NLP** | Natural Language Processing |
| **k-NN** | K-nearest neighbor |
| **SMOTE** | Synthetic Minority Oversampling Technique |
| **POS** | Part of Speech |
| **LIWC** | Linguistic Inquiry and Word Count |
| **SAX** | Symbolic Aggregate approXimation |
| **MSA** | Multiple Sequence Alignment |
| **SVC** | Support Vector Classification |
| **RBF** | Radial Basis Function |
| **PCA** | Principal Component Analysis |
| **ANOVA** | Analysis of Variance |

# Chapter 1

# Introduction

This chapter includes a general overview of the background of the research topic, with the research question, research objective and the scope of the research. A detailed documentation of the background methodologies used in the paper is also included in the chapter.

## 1.1 Background

Nowadays, With the developing of technology and the increasing popularity of a wide range of different online social media, in 2018 over 4 billion people worldwide have access to the internet and over 3 billion of them are active on social media (Social, 2018). Social media enables the user to develop connections, to share details of life as well as to publish pictures and videos.

Now with the rapidly increasing accessibility to the internet, children and adolescents are increasingly involved in the usage of social media. As suggested by (Livingstone, Haddon, Görzig, & Ólafsson, 2011), according to a survey on 25,142 children in the EU at the age of 9 to 16 who uses the internet, the average daily usage time is 88 minutes, with 60% of the children goes online daily. 49% of all children who uses the internet states that the usages are carried out in their privacy, without being supervised by their parents. However, the vast developing of the online network not only

brings people from different part of the world into connection, but at the same time, it also provides a platform and opportunity for criminals to approaches the potential victims through social media. For all the children participated in the survey, 34% stated that they had added people they never met face to face onto their social media friend list, where 15% had sent personal information including photos and videos to strangers and 9% agreed to meet with people they met online in person.

The social network does not only bring convenience to the world, with the lack of monitoring and ethical standards, social media have also been used as the platform for bullying, assault and harassment. (Esposito, 1998) suggested that online sexual content exists from nearly the start of the internet, while online sexual offending followed shortly after. Based on the survey by (Livingstone et al., 2011), 1 in 8 children encountered sexual images and received sexually related messages. Yet even though the exposure of children to harmful content and individuals has become a common issue, the prevention of such event and conviction of the criminals remain a global issue. As stated by (Jeney, 2015) the media platforms can be used to solicit children which can lead to sexual exploitation, while at the same time, the virtual community allows the offenders to remain anonymous or even create fake virtual identities which allow them to approach the children without being suspicious.

In this paper, the focus is on identifying the offenders that harass the children with sexually related messages. The offenders in this particular scenario are referred to as online predator, where the definition is given by (Morris, 2013) as an adult engages in conversations with the underage individual (under the age of 18), the adult also introduce or encourage intimate conversation. (Cheong, Jensen, Gunadóttir, Bae, & Togelius, 2015) suggests that the predators may initiate sexually suggestive language, who may also have the attempt to gain physical access to the victim to meet the victim in person.

The traditional way of identifying online predators is by having trained volunteers

posing as adolescents in the chat rooms waiting to be approached by predators, have conversations with the predators and eventually bring them into conviction. The approach is highly dependent on human force and can be extremely time-consuming. As a result, automated tools need to be constructed for the detection of online sexual predators. In the paper, a classification system for predator detection is proposed, as well as the analysis of predator features and behaviours which will lead to likely future works in the related area.

## 1.2 Research Project/problem

Due to ethical reasons, the conversations between predators and the victims are rarely published online. The PAN 2012 Sexual Predator Identification dataset (see section 3.1) is the only benchmark dataset contains conversations between predators and victims available online in English (Peersman, 2018). Previous researches had been carried out to build predator identification models using the dataset. Due to the limitation on the number of predatory conversations published online, the dataset is extremely biased with less than 1% of the authors in the dataset being predators results in a significantly unbalanced dataset. A better classification performance might be achieved by re-balancing the dataset.

Researches have been carried out on author profiling, i.e. the identification of an author's age and gender group based on blogs, posts from social media etc (Rangel Pardo et al., 2015). For the task of predator identification, due to the specific age group of the victims (adolescents), the age of the author might be a useful feature to identify the victims, which may lead to the identification of the corresponding predators.

The performance of the predator identification model might be improved by re-balancing the unbalanced dataset. The age of the authors may also be extracted from the conversations, which can be used as features to build a better performing classifier.

## 1.3 Research Objectives

The aim of the research is to evaluate the effect of re-balancing the training set on the performance of the predator identification model, as well as determine whether the age of the authors can be extracted from the dataset and contribute to the identification of predators.

The main objectives of the paper include:

1. To build a SVM classifier for predator identification, the optimum performance of the classifier is achieved by testing various parameter setting and kernel options.

2. To test whether re-balancing an unbalanced training set can improve the performance of the model. The original training set is randomly downsampled into training sets with different ratio between the samples of the majority and the minority class. Experiments are carried out to test the performance of the classifiers trained on the training sets.

3. To test whether the age profile of the authors in the dataset can be extracted and hence be used to train the predator identificaiton classifier. Experiments are carried out to construct an age identification model, the performance of the model is tested and evaluated on the PAN 2012 Sexual Predator Identification dataset.

## 1.4 Research Methodologies

The experiment carried out for the dissertation consists of secondary research, as the dataset used in the paper is obtained from the PAN12 competition. The objectives of the research are quantitative, as numerical measurements and quantification are required in the research process. The form of the research is constructive, where a new construct is being developed. The experiment goes through the process of theory, hypothesis, observation and then confirmation, hence the research is deductive.

## 1.5 Scope and Limitations

The research is limited to the detection of English speaking predators collected from Perverted Justice (PJ) [1]. PJ is an American website with conversations from convicted predators from USA from the year of 2004, hence the predatory conversations used in the research are collected between the year of 2004 and 2012. Due to the nature of the website, the conversations are between predators and pseudo-victims (adult volunteers posing as adolescents), conversations between predators and victims (victims are adolescents) are not obtained. The dataset used for the age identification classifier is collected in the year 2006 from online age-specific chat room, which is within the time the predatory conversations are collected.

## 1.6 Document Outline

The paper is organised as the following:

- **Chapter two: Review of existing literature** will focus on reviewing the previous researches carried out on text classification including the general methodology and pre-processing of the text document. Researches on the effect of re-balancing unbalanced datasets will also be included in the chapter. A detailed overview of previous literature on the detection of online criminals and sexual predators will be given. Researches carried out on author age profile identification will also be addressed in the chapter.

- **Chapter three: Experiment design and methodology** will outline a detailed design of the experiments carried out for the research. The datasets and terminologies used in the paper are given, along with a detailed explanation of the Python packages and functions used for the research. The evaluation matrix and statistical significance test choices are also explained in the chapter. The experiment design section in the chapter will clarify the experiments carried out

[1] http://www.perverted-justice.com/

to build the predator identification model, to evaluate the effect of re-balancing an unbalanced dataset as well as to build the age group identification model.

- **Chapter four: Results, evaluation and discussion** will provide the results obtained from the experiments. The performance of classifiers trained with dataset with different scales of unbalance are compared and evaluated. The performance of the age prediction model is also evaluated in the chapter, and compared with the results from the literature review. The overall performance of the predator identification model would be provided and analysed.

- **Chapter five: Conclusion** will review the overall experiments carried out in the dissertation and the results obtained. Potential improvement in the model and future areas of investigations are also outlined.

# Chapter 2

# Review of existing literature

The following chapter contains an overview of the previous literature, the chapter is divided into four sections:

1. Researches on text classification.
2. Literature on unbalanced datasets.
3. Previous researches on online crime and predator identification.
4. Previous work on author age profiling.

## 2.1 Text Classification

(Khan, Baharudin, Lee, & Khan, 2010) states that the goal of text mining is to extract useful information from textual resources and hence accomplish tasks like retrieval, classification and summarisation. For the case of dealing with real-life messages or documentation, the idea of Natural Language Processing (NLP) is introduced, which is defined by (Khan et al., 2010) as to gain a better understanding of natural language by using computational force and representing the textual documents semantically for an improved classification performance.

(Joachims, 1999) stated that the goal of text classification is to assign textual doc-

uments into predefined semantic categories. By introducing machine learning algorithms (supervised, unsupervised and semi-supervised) the objective is to achieve automatic category assignment by constructing classification models from the training set. For the paper only supervised classification techniques are considered due to for the majority of the text classification scenarios, the documents are required to be assigned to predefined categories based on a training set of labelled documents. (Chau & Chen, 2008) suggested a list of major for text classification using machine learning methods including K-nearest neighbor (k-NN), support vector machine (SVM), neural network, decisiou tree etc.

SVM is a traditional classification method, it performs effectively in text classification tasks, due to the large dimensionality of the text data (Shah & Patel, 2016). SVM has also been proven to work effectively for high-skew text classification tasks (Desmet & Hoste, 2014). (Yu, Ho, Juan, & Lin, 2013) states that for the cases of large and sparse datasets with high dimension, the linear kernel for SVM would be the optimum option. As a result, most of the classifiers built in the paper used the SVM algorithm, a k-NN model is also constructed to compare with the performance of the SVM classifiers.

The k-NN algorithm is explained by (Yang, Liu, et al., 1999) as for a given test data point, the class of the entity is found by finding its k nearest neighbor among training data points and to obtain a category by weighing the categories of the training data points. The similarity score between the test data point and the neighbors is used to weigh the categories of the training data points by accumulating the scores of the same category. In other word, the test data point is assigned to a category if among the k nearest training data points it is the most frequently obtained category. The decision rule of k-NN is defined by (Tan, 2005) as:

$$score(d, c_{\mathrm{i}}) = \sum_{d_{\mathrm{j}} \in KNN(d)} Sim(d, d_{\mathrm{j}}) \sigma(d_{\mathrm{j}}, c_{\mathrm{i}}) \tag{2.1}$$

where KNN(d) is the set of k nearest neighbors among the training data points to the test data point d. $\sigma(d_j, c_i)$ is the classification of the data point $d_j$ with respect to the category $c_i$. In another word, $\sigma(d_j, c_i) = 1$ if $d_j \in c_i$, otherwise, $\sigma(d_j, c_i) = 0$. The test data point d is then assigned to the class with the highest score.

K-NN suffers from the curse of dimensionality and requires a large amount of training samples to successfully classify samples with many features, which might results in overfitting (Hartmann, Huppertz, Schamp, & Heitmann, 2019).

### 2.1.1 Support Vector Machines

The Support Vector Machine (SVM) is a supervised classification method which classifies unknown instanced based on models built from labelled objects introduced by (Boser, Guyon, & Vapnik, 1992). As stated by (Noble, 2006), due to its ability to work with high-dimensional data, SVM has been used in various fields, including handwriting recognition, fraudulent detection etc.

(Vapnik, 2013) stated SVM as an approach that maps the input vectors into a high dimensional feature space through some nonlinear mapping. An optimal separating hyperplane is constructed in this space. As suggested by (Sun, Lim, & Liu, 2009) the optimally separating hyperplane performs as the decision surface, which finds the largest margin between the positive training example from the negative ones.

(Hastie, Tibshirani, & Friedman, 2009) illustrates the method of learning the mathematical representation of the hyperplane. Given the training data consists of pairs ($x_i$,$y_i$), with $x_i$ represents the feature vector of the $i$th training point, and $y_i \in \{-1, 1\}$ represents the label of the $i$th training point. The hyperplane is defined by

$$x : f(x) = x^T\beta + \beta_0 = 0, \tag{2.2}$$

where $\beta$ is a unit vector perpendicular to the hyperplane given $\|\beta\| = 1$, $\beta_0$ gives the position of the hyperplane. The hyperplane can be optimised to create the biggest

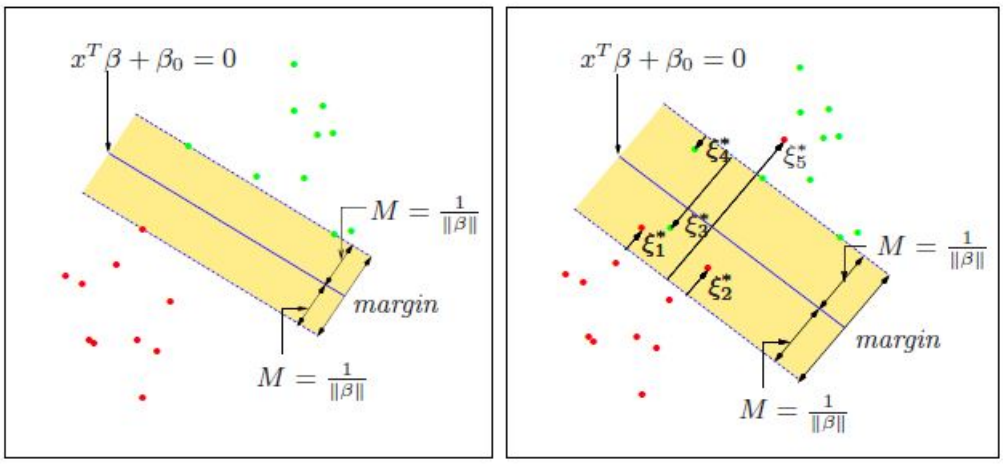


Figure 2.1: Margin and hyperplane expression for SVM

margin between class 1 and -1 training points. As shown in figure 2.1 the decision boundary on either side of the hyperplane is M unit away,where $M = 1/\|\beta\|$. Hence the margin 2M units wide, where $2M = 2/\|\beta\|$. The expression of the optimum hyperplane can be found by maximising the margin, or on the other hand to minimise $\|\beta\|$. The problem can be expressed as

$$\min_{\beta,\beta_0} \|\beta\|$$
$$\text{subject to } y_i({x_i}^T\beta + \beta_0) \geq 1, i = 1, ..., N, \tag{2.3}$$

After the equation for the hyperplane ($f(x) = x^T\beta + \beta_0$) has been learned, the sign of the score ($f(x)$) is used to classify the labels of the testing document. The default threshold of 0 is normally used for the SVM classifiers, As a result a score of $f(x) \geq 0$ are classified as positive while a score of $f(x) < 0$ are classified as negative. By altering the threshold of the SVM classifier the prediction outcome of the document using the classifier can be modified.

### 2.1.2 SVM Parameters in Computing

#### C and Gamma Parameters

During the computing stage of the SVM classifier, in order to achieve optimum performance, several parameters can be tuned. As stated by (Lameski, Zdravevski, Mingov,

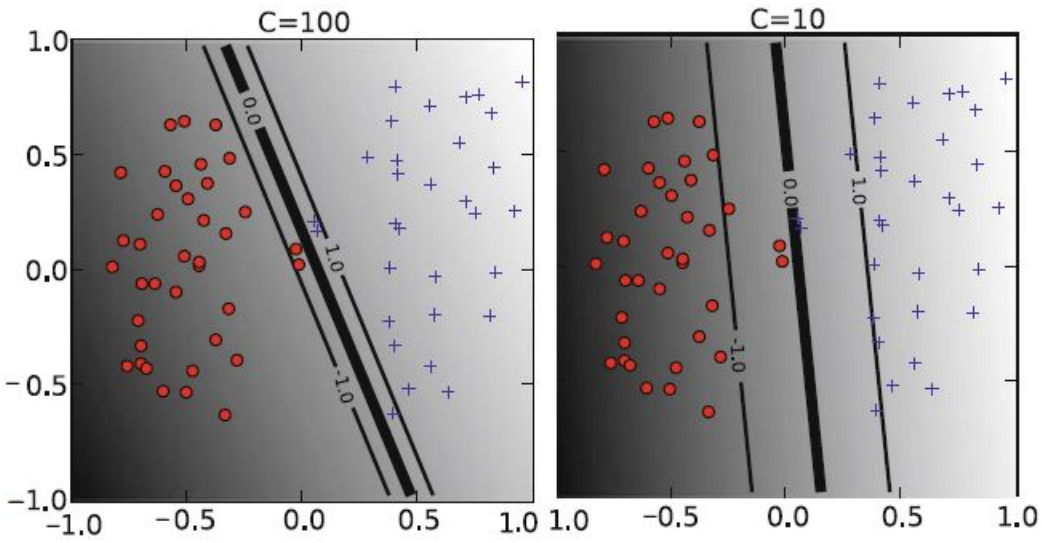


Figure 2.2: The effect of C parameter on the svm model decision boundary with C=100 and C=10.

& Kulakov, 2015) the parameters C and gamma are provided as inputs to the SVM, which alters the process when finding the ideal hyperplane.

(Ben-Hur & Weston, 2010) suggests the C parameter defines the tolerance on margin errors, as shown in figure 2.2, where the greater the C value (left figure), a larger the penalty is assigned to errors. As a result by increasing the value of C, less points are allowed in the error margin, while a smaller value of C (right figure) allows the margin error to be larger and hence more points in the margin. The gamma parameter influences the flexibility of the line of the hyperplane, where for smaller gamma values the separation line of the hyperplane approaches linearity, alternatively for larger gamma values the line becomes more curved. As shown in figure 2.3, when gamma = 0.1 (top left figure) the decision boundary is nearly linear, whereas the gamma value increases, the decision boundary becomes more flexible, with the boundary closer fitted to the sample points. When gamma = 100 (bottom right figure, the decision boundary becomes closely wrapped around the sample points, which results in overfitting of the model.

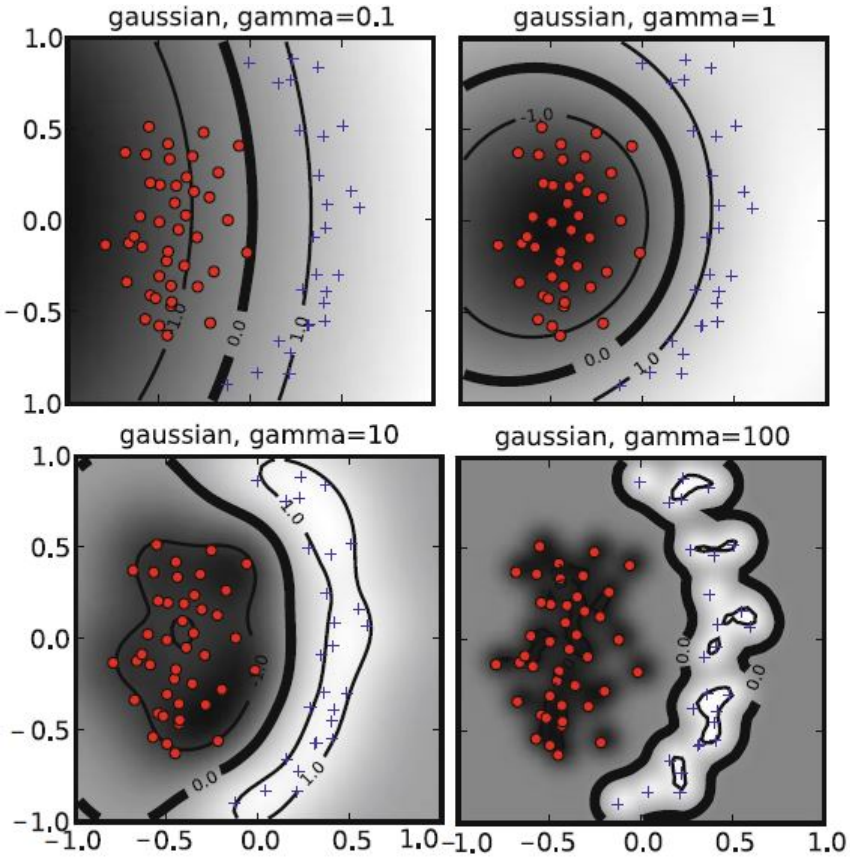


Figure 2.3: The effect of gamma parameter on the svm model decision boundary with gamma=0.1, 1, 10 and 100

When training a SVM classifier on unbalanced dataset, extra weight can be applied to the minority class to put more emphasis on correctly classify points belonging to such class.

#### Kernels in SVMs

(Leslie, Eskin, & Noble, 2001) states the optimisation of a SVM is equivalent to solving a dual quadratic programming problem, the kernel technique is introduced to transfer the input data to format required for the classification process. Different types of kernel have been introduced, including polynomial, linear, radial basis function (RBF) and sigmoid. For training instances $(x_{\mathrm{i}}, x_{\mathrm{j}})$, the kernels can be written in the following expressions according to (Hussain, Wajid, Elzaart, & Berbar, 2011) and (Wang & Hu, 2005):

(i) Linear kernel:

$$K(x_{\mathrm{i}}, x_{\mathrm{j}}) = (x_{\mathrm{i}}, x_{\mathrm{j}}) \tag{2.4}$$

(ii) Polynomial kernel:

$$K(x_{\mathrm{i}}, x_{\mathrm{j}}) = ((x_{\mathrm{i}}, x_{\mathrm{j}}) + p)^d, \quad d \in N, p > 0 \tag{2.5}$$

(iii) RBF kernel:

$$K(x_{\mathrm{i}}, x_{\mathrm{j}}) = exp(-\gamma \|x_{\mathrm{i}} - x_{\mathrm{j}}\|^2, \quad \gamma > 0 \tag{2.6}$$

(iv) Sigmoid kernel:

$$K(x_{\mathrm{i}}, x_{\mathrm{j}}) = \tanh(\gamma(x_{\mathrm{i}}, x_{\mathrm{j}}) + \theta), \quad \gamma > 0, \theta > 0 \tag{2.7}$$

where d, p, $\gamma$ and $\theta$ are variables that can be adjusted based on the data and the classification task.

### 2.1.3 N-Grams

#### Character N-Gram

An character N-gram is defined by (Cavnar, Trenkle, et al., 1994) as an N-character slice of a longer string. Typically the string is sliced into a set of overlapping units of n

characters, where n is an integer. The blank spaces in the string are also considered as a character represented using '_'. For instance, the word 'document' can be decomposed as the following N-grams:

1. Uni-gram (n=1): d,o,c,u,m,e,n,t
2. Bi-gram (n=2): do,oc,cu,um,me,en,nt
3. Tri-gram (n=3): doc,ocu,cum,ume,men,ent

use of character n-grams takes in to consideration the sequence of the words in the documents, instead of simply tokenize the items in the strings and carry out analysis with individual units.

**Word N-Gram**

The word N-gram is similar to the character N-gram, while instead of separating each character in the string, the string is separated into individual words. For instance, the string 'word ngram bags of words' can be decomposed as the following N-grams:

1. Uni-gram (n=1): word, ngram, bags, of, words
2. Bi-gram (n=2): word ngram, ngram bags, bag of, of words
3. Tri-gram (n=3): word ngram bags, ngram bags of, bags of words

The use of n-grams takes in to consideration the sequence of the words in the documents, instead of simply tokenize the items in the strings and carry out analysis with individual units.

### 2.1.4 TF-IDF

The theory of N-grams has been explained in the previous section, however as stated by (Arumugam & Shanmugamani, 2018), the major drawbacks of the approach is

that the vectors extracted from words present in the document are given equal weight. Different approaches can be used by using the weighted average of the vectors, one of the ways is to use the $tf - idf$ weighing, which can be expressed using the following equation:

$$tf - idf(w, d, D) = tf(w, d) \times idf(w, D) \tag{2.8}$$

where w represents the term in the documents, d represents a document and D represents the entire text corpus. As a result, $tf(w, d)$ is the number of times(frequency) the term w appears in the document d, and $idf(w, D)$ can be expressed as:

$$idf(w, D) = log \frac{1 + |D|}{1 + df(d, w)} \tag{2.9}$$

where $|D|$ represents the total number of documents in the corpus, $df(d, w)$ gives the number of documents the term w appears in. The equation can be interpreted as the logarithm of the total number of documents in the corpus divided by the the number of documents a term appears across the corpus.

(Arumugam & Shanmugamani, 2018) states the value of the $tf - idf$ weight score follows the following rules:

1. The score would be the highest if a term w is found in some but not all documents in the corpus.

2. The score would be lower if the term w appears in too little or too many documents in the corpus

3. The score would be the lowest if the term w appears in all of the documents with multiple appearances per document

As a result, the $tf - idf$ approach can be combined with the n-grams to get a weighted representation of the term in the documents while the sequences of words are taken into consideration.

### 2.1.5 Text Preprocessing

(Onan, 2018) states the using appropriate feature set in machine learning based text classification is essential for the optimum model to be generated. However, for text documents, as a format of unstructured data, techniques such as preprocessing need to be carried out to reduce the complexity of the document.

(Khan et al., 2010) carried out preprocessing through tokenization, stop words removal and stemming for the first step of the preprocessing stage. Where tokenization is to break the document from its original string format into a list of tokens, stop words removal removes a list of words that occurr frequently in the document with insignificant contribution to the content of the document and stemming transforms different words into standard form (eg: running to run, connection to connect etc.). (Vijayarani, Ilamathi, & Nithya, 2015) suggested multiple methods for text documents preprocessing including stop words removal, stemming as well as using tf-idf to find the importance of a word in the collection of document, which can hence be used for further stop words removing. Although similar theories and approaches are followed by the majority of the preprocessing stage, the methodologies can be different based on the nature of the data used and the objective of the research, further information will be included in section 2.2 when the research is focused on various types of real-life text documents (email, blog, messages etc).

## 2.2 Detecting cybercrime and predator identification

With the properties of text classification algorithms explained in the previous sections, this section gives an overview of previous studies that used text classification methods for cybercrime identification and previous works on predator identification.

(Van Hee et al., 2015) used text classification algorithms for the detection of cy-

berbullying events (the aggressive, intentional offend which is carried out through an electronic platform). Features including the word unigram and bigram, character trigram and sentiment features are used for the classification algorithms. The sentiment feature represents the polarity of the lexicon words, can be positive, negative or neutral. The features are classified using SVM algorithm to find posts containing cyberbullying content. In the paper the cyberbully identification model achieved an overall f-score of 55.39, however, the sentiment features contributed extremely poorly to the performance, with a f-score of 6.35 obtained when using the sentiment feature in isolation in comparison to the 47.94 achieved when using the word unigrams in isolation.

(Dinakar, Reichart, & Lieberman, 2011) focused on the detection of textual cyberbullying using a corpus consist of YouTube video comments. The data is preprocessed by removing stop words, stemming and removing of a sequence of characters that are not important. Based on the nature of the corpus, the unimportant sequence of characters includes user id, the repeating of characters in a word ('lolllllll') etc. Tree based J48 and SVM are used for the classification tasks, where features used including tf-idf weighted uni-grams, part of speech (POS) bigram tag which indicates the grammatical category of the tokens in the text corpus, as well as a list of modified Ortony lexicon which contains only words with negative affect. A 66.7% accuracy is obtained by the SVM classifier and 61% accuracy obtained for the Tree based J48 classifier.

Out of a range of different types of cybercrime, the paper is focused on the detection of sexual cyber offenders which is also defined as predators. A notable early example on the topic is from (Pendar, 2007), which used unigram, bigrams and trigrams for the classification using SVM and k-NN classifier to classify the predators in the chat corpus. For the study, all the conversations used in the dataset belongs to the positive category, as all the conversations in the corpus include at least one predator. The corpus contains real online chats, which contains different vocabulary from formal textual documents. As a result, the preprocessing stage of the corpus included the

removal of a stop list of the 79 most frequent word types in the corpus, neither stemming not mis-spelling are corrected in the paper to retain the non-standard format of the language used in the corpus. For the instance of repeating letters (eg: noooooo), only one letter of the repeating letter was left. The F-score is used for the evaluation of the performance of the models, where an F-score of 0.91 with SVM and 0.94 with k-NN is obtained for the trigrams, however, a low F-score of between 0.42 and 0.58 is obtained for unigram and bigram.

(Kontostathis, 2009) also proposed a method for separating the predator and victim. They proposed their method the 'Chatcoder', which is an application that separates the predatory activity into different stages (approach, isolation etc.), the messages in the conversations are categorised using a simple dictionary of words and phrases. Another part of the experiment constructed a J48 classifier for the categorisation between predators and victims, a C4.5 decision tree is built using reduced-error pruning, an accuracy of 60% is obtained.

In year 2012 the PAN competition was then held with the sexual predator identification task, a range of approaches are published with the results. (Villatoro-Tello, Juárez-González, Escalante, Montes-y Gómez, & Pineda, 2012) used a two step approach, where for the first stage, the suspicious conversations are identified, and the second stage the predators are separated from the victims. Tello's approach does not contain any preprocessing process due to the nature of the chat corpus, the intentional misspelt words may contain useful contextual information. Although no preprocessing is carried out, a pre-filtering stage is used to reduce the computational cost for the classification task. For the filtering stage, conversation with only one author, conversations that have less than 6 interventions per author and conversations that have long sequences of characters with no meanings. For the filtered data corpus, the size of the training corpus is significantly smaller, with 90.2% of the conversations removed, however, 12 out of 148 predators are also removed, which makes the approach impossible for identifying 100% of the predators. An F-score of 0.9346 is achieved using the

approach, which is ranked first in the competition according to (Inches & Crestani, 2012). Tello also applied dimensionality reduction to the features used to train the model, a decrease of the model performance is obtained after reducing the number of features used.

(Parapar, Losada, & Barreiro, 2012) took a different approach for the preprocessing stage for the PAN12 competition. The vocabularies appeared in less than or equal to 10 conversations are removed, terms with character size greater than 20 are removed, the bigrams and trigrams that appeared in less than or equal to 3 conversations are removed and the N-grams with character size greater or equal to 40 are removed. SVM is used for the classification process using features including tf-idf features, LWIC features and chat-based features such as the percentage of messages sent by an author (across conversations participated by the author), the average time of the day the author chats etc. Linguistic Inquiry and Word Count (LIWC) is a text analysis software that gives the psychological aspects of natural language, the output contains 80 LWIC features including the psychological aspect (affect, cognition etc), personal concern categories etc. An F-score of 0.8691 is achieved for the classification stage.

(Morris, 2013) constructed the classifier using a slightly different approach. A wider range of different features are extracted from the data set and used for the classification model, including the lexical features such as unigrams, bigrams as well as extracting the emoticons with their sentiments (happy, sad, etc). Behavioural features are also used including the number of messages sent by an author in the corpus, the number of conversations an author participated in the corpus, as well as features describing the initiative, attentiveness and conversation dominance of the author. An SVM model is trained using the features extracted, following with a postprocessing stage. The model achieved an F-score of 0.8652, where the lexical features and the postprocessing stage contributed most to the performance of the model, the behavioural features failed to contribute significantly to the model's performance.

(Hidalgo & Díaz, 2012) approached the PAN12 competition by using a previously existing knowledge based system for predator detection in Spanish. The system, similar to the Charcoder, identifies the different stages of predatory behaviour. The content of the system is automatically translated to English using Google Translator without manual correction of potential mis-translations. A Naïve Bayes classifier is trained with the output from the knowledge based system combined with other linguistic features including the number of uppercase letters, the number of words etc. An F-score of only 0.775 is achieved using the model, the not ideal performance is stated to be due to the automatic translation of the system from Spanish to English, or the lack of lexical features that should be used for building the classifier (unigrams, bigrams etc.).

After the PAN12 competition, the research of automatic detection of online predators is still a topic that is widely interested, multiple different approaches have been made since.(Cheong et al., 2015) focused on the detection of predatory behaviour in game chats, the research is carried out on chat logs from the community of a specific game, as a result the nature of the chat corpus can be slightly different due to the uniqueness of the chatting style of the game. However, similar features are extracted from the data, the lexical features and the behavioural features. For the lexical features, the sentiment score is also included using the AFINN-111 word list, which is a dictionary contains the labels and scores of the sentiment words. The data corpus is preprocessed by manually eliminating the messages that clearly contains no predatory behaviour. The Multilayer Perceptron algorithm is used to build classification models, with an F-score of 0.86 achieved when testing on data collected from the same game chat. Cheong also tested the model on the PAN12 test corpus, an F-score of only 0.12 is achieved, the poor performance is stated to be from the different in size between the training and test sets, as the test set is significantly larger.

(Potha, Maragoudakis, & Lyras, 2016) suggested a way of using a method inspired by computational biology algorithms. The patterns within the predatory stage are converted into strings in Symbolic Aggregate approXimation (SAX) representation,

which then passes the strings to a Multiple Sequence Alignment (MSA) algorithms which synthesises the patterns within the sequence. The dataset is collected from Perverted-Justice(PJ) [1], which contains convicted conversation between predators and victims, the lines sent by predators are labelled with numeric values indicating the degree of predatory involved in the line. The numeric values are then transferred into a sequence of time series data, then into the sequence using the SAX method. A correlation coefficient of 0.989 is achieved.

## 2.3 Unbalanced Dataset

(Chawla, Japkowicz, & Kotcz, 2004) stated the class unbalance problem occurs when, in a classification problem, there are many more instances of some classes than others. According to (Provost, 2000) most of the machine learning algorithms follow the following assumptions:

(i) The goal for the model optimisation is achieved by maximising the accuracy,

(ii) The classifier will be used for dataset with the same class distribution as the training set.

As a result standard classification algorithms tend to learn the class with the larger population better while ignoring the smaller class. Specifically for SVM, as stated by (Akbani, Kwek, & Japkowicz, 2004), the performance of SVM can be affected by unbalanced training set when negative training entities heavily outnumber the training entities of the positive class. However, as stated by (Han, Wang, & Mao, 2005), for most of the real-world domains, the correct classification of the minority class is more critical than the majority class. Thus, multiple approaches are introduced to improve the prediction for the minority class.

As suggested by (Sun et al., 2009), a number of ways have been proposed addressing

[1] http://www.perverted-justice.com/

the unbalanced dataset, including:
(i) Up-sampling the minority class by randomly generating entities obtaining similar feature as the original entities or by duplicating the entities to re-balance the training set. One of the most popular up-sampling method is the Synthetic Minority Oversampling Technique (SMOTE) method proposed by (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), which creates new points for the minority class by finding a point belonging to the class and its nearest neighbors, the new point is placed randomly in between the instance and its neighbors. The drawback for up-sampling is it can make the already large dataset even larger and leads to a increase in computational time required.
(ii) Down-sampling the majority class by randomly choosing less entities belonging to the majority class to create a more balanced training set.
(iii) Allocate different weighting to entities of the minority and majority classes in the training set. With more weight allocated to entities of the minority class.
(iv) To alter the thresholds of the classifier to balance the precision and recall of the predicted set.

Due to the nature of the training set used in the paper, with 2,016 suspicious conversations (The conversation that involves at least one predator as an author) and 64,911 non-suspicious conversations. As stated by (Akbani et al., 2004), for real-life instances similar to the identification of credit card fraud, the probability of being a fraud is significantly lower than being a valid transaction. Thus assumptions are likely to be made by the mechanism to classify the instance as not a fraud. Similarly, in the case of detecting predators, using the original training set, instances are more likely to be classified as non-suspicious conversation based on the default assumption of the model.

Several participants in the PAN12 competition addressed the unbalanced dataset in their approach, Parapar allocated additional weighting to the predator class, while Morris performed post thresholding to the model. For this paper, the down-sampling method is used to generate a collection of training sets with different ratio between the suspicious conversations and non-suspicious conversations to test the effect of im-

balanced training sets have on the model.

## 2.4 Age identification

The objective of the paper is to detect the predators in the chat logs, based on the definition of predators, the victims that involved in conversations with the predators must be under the age of 18, thus it might be possible to identify the predators by taking the approach of identifying the victims first, using the age of the person as a feature used for the classification. Age prediction from chat data has been carried out in previous studies, (Tam & Martell, 2009) carried out age prediction between chat corpus of authors in the age groups of teens, 20s, 30s, 40s and 50s. SVM and Naïve Bayes classifiers are used with n-grams to constructed the models for age prediction. The SVM classifier performed better than the Naïve Bayes classifier in general, an F-score of 0.996 is achieved when distinguishing teens from adults using a SVM classifier with tri-gram.

(Pentel, 2015) also carried out age identification using short texts from authors of groups of children and teens less than the age of 16 and adults over the age of 20. Multiple algorithms are used to build the classifiers including logistic regression, SVM, k-NN and Naïve Bayes classifier. Only readability features are used as features for the training corpus, including the average number of characters in a word, the average number of words in a sentence etc. The best classification result is achieved by the SVM classifier with an f-score of 0.94.

(Peersman, 2018) carried out an experiment having a group of adults over the age of 25 posing as teenagers participate in one on one conversation with adolescents between age 12 and 14. The messages sent by the adult volunteers are collected and classified using an age prediction classifier between two classes adolescents and adults. The experiment result indicated that all the adolescents participated failed to identify their chat partner's real age group, as 100% of the adolescents identified their partner

| Author | Remove most frequent | Remove least frequent | Remove large words | Conversation pre-filter |
|---|---|---|---|---|
| Tello | × | × | × | ✓ |
| Parapar | ✓ | ✓ | ✓ | × |
| Morris | × | × | × | × |
| Pendar | ✓ | × | × | × |

Table 2.1: List of preprocessing techniques used from previous literature

as younger than 16. However, all of the adult participants are able to be identified as adults using the machine learning classifier.

## 2.5 Summary

Text classification algorithms are reviewed in the chapter, with SVM chosen as the main algorithms to be used for the experiments in the paper. Online predator detection have been carried out in many previous researches. The pre-processing techniques used in the researches are illustrated in table 2.1, for most of the researches stop words are not removed from the training set, stemming is also not carried out. Some of the researches removed the n most frequently appeared word and n least frequently appeared word, for other instances, conversations are removed from the training set based on the number of authors participated in the conversation and the number of total messages in the conversation. For some instances, the training set is not pre-processed.

Various text classification algorithms are used in the approaches, including k-NN, decision tree, neural networks and SVM, where SVM is the most common approach.

### 2.5.1 Gaps in literature

The problem with the unbalanced dataset used for the PAN12 predator identification has been addressed in previous work through allocating additional weighting to the

minority class and post-thresholding of the classifier. However, the performance of downsampling the original dataset is not explored in the previous literature.

The author age prediction, although have been proven successful in multiple researches, has not been implemented in the predator identification tasks. It is proposed for the research, to extract the age profile of the authors as part of the features used for the classifiers.

# Chapter 3

# Experiment design and methodology

This chapter outlines a detailed design of the experiments carried out for the research. A full summary of the dataset and terminologies used in the paper is provided along with the methods to deal with the imbalanced dataset and extract useful features from the dataset. The classification techniques are also illustrated including the Python packages used to construct the classifiers. A detailed explanation of the evaluation metric, dimension reduction and statistical significance test is also included in the chapter. A two stage classification method is introduced in the chapter, with suspicious conversations identified in the first stage and the predators identified in the second iteration. An age prediction model is also designed in the chapter to identify the age group of the authors.

## 3.1 Dataset

The dataset used for the paper is from the PAN12 competition which was held in conjunction with CLEF 2012 (Conference and Labs of the Evaluation Forum). According to (Inches & Crestani, 2012) the PAN12 dataset consists of chatlogs including real predators collected from Perverted Justice (PJ) [1], sexually related chatlogs between

[1] http://www.perverted-justice.com/

adults from the Omegle repository [2] which is an online website allows the user to chat to random strangers, chatlogs about generic topics are also included in the dataset which are IRC logs drawn from [3] and [4].

The PJ Foundation is an organisation based in the United States of America, where volunteers are trained to pose as adolescents in online chats to be approached by predators. The conversations posted on the PJ website are collected from convicted cases, and contains the online chatlog between the predator and the volunteer from the initial conversation until the reveal of obvious sexual criminal attempt from the predator. There are currently over 600 chatlog convictions on the PJ website, where 142 of the predators are included in the training dataset out of 97,689 of the total authors, for the testing dataset, 254 predators are included out of the 10,000 authors. The dataset consists of conversations each with a unique conversation id, with the author id and the time of the day given for each message line in the conversation.

The NPS Chat Corpus[5] from (Forsythand & Martell, 2007) is used to train the age prediction model. The NPS Chat Corpus consists of messages collected at different time in 2006 from age-specific chat rooms, including teens, 20s, 30s, 40s and adults. Each file downloaded consists of approximately 700 chat logs from a specific age group. The messages in the chat corpus are labelled with unique author ids and the age group of the author. The NPS Chat Corpus is only used for the age feature extraction experiment in section 3.9.3.

## 3.2 Terminology

**Author** The participant in a conversation, each author is labelled with a unique author id.

[2] http://www.omegle.com/
[3] http://www.irclog.org
[4] http://krijnhoetmer.nl/irc-logs
[5] http://faculty.nps.edu/cmartell/NPSChat.htm

**Predator** The definition of predator has been explained in section 1.1.1 as the adult engages in intimate conversation with under-aged individuals. For the purpose of the study, the predators are defined based on the list of predator author id provided by the dataset.

**Victim** A victim is the author participate in a conversation with the predator.

**Message** A message is a string of text sent by an author, with the timestamp of the message sent provided in the corpus.

**Conversation** A conversation is a set of messages sent from one or more authors, with the messages arranged in successive order. The time gap between consecutive messages is required to be less than 25 minutes to be in the same conversation.

## 3.3 SVM in Python

Python is used to construct the SVM classifiers used in the paper, the scikit-learn package from (Pedregosa et al., 2011) is used. It is a toolkit built on NumPy, SciPy and matplotlib, can be used for completing machine learning tasks in Python.

### TF-IDF Vectorizer

The N-Gram tf-idf in the paper is carried out using the sklearn.feature_extraction.text. TfidfVectorizer, which converts the training set from raw document to tf-idf feature matrix and used for building a SVM classifier. The parameter of the vectorizer can be altered to modify the N-Gram used for the vectorizer, with the choice of carrying out character N-Gram or word N-Gram, the n values used for the N-Gram can also be modified as N-Gram with n in the range(min_n, max_n).

### K-Fold Cross Validation

When selecting the optimum model in the training stage, the K-fold cross validation method is used. An explanation of the cross validation method is given by (Bengio & Grandvalet, 2004). The technique repeats the training algorithm K times by diving the training set into training and validation set, with $\frac{1}{K}$ of the training set used for validation each iteration, while the rest used for the training of the sample. (Kohavi et al., 1995) states for the K times the training process is repeated, the validation set are mutually exclusive, hence each instance in the training set is used once and only once as the validation entity. (Stone, 1974) states that although the repeating in the training process reduces the computational efficiency, the variance of the estimate can be lowered using the technique.

When computing the SVM classifier, the stratified K-fold cross validator is used, where the proportion between the class are preserved to be the same for each fold and the same as the original training set stated by (Diamantidis, Karlis, & Giakoumakis, 2000). The sklearn.model_selection.StratifiedKFold function is used to carry out the cross validation task, a 10 fold cross validation is used, the training set is also shuffled for instances of each class is splitted into groups.

### SVM Kernel Functions

In sklearn a SVM classifier can be constructed using the sklearn.svm.SVC function, where SVC stands for support vector classification. The classifier is derived from LIBSVM, which is a software used for completing machine learning tasks including support vector classification, regression and distribution estimation as stated by (Chang, 2011). The type of kernel used for generating the SVM classifier can be altered by changing the 'kernel' parameter in the sklearn.svm.SVC function, kernels including 'linear', 'poly', 'sigmoid' and 'rbf' can be used for the function.

Similar to the sklearn.svm.SVC function with kernel='linear', function sklearn.svm. LinearSVC function can be used. According to (Fan, Chang, Hsieh, Wang, & Lin,

2008) the function is based on LIBLINEAR instead of LIMSVM, for some large dataset, the classifier can achieve similar performance with and without the use of nonlinear mappings. For the sklearn.svm.LinearSVC function, kernels are not required when training the linear classifier, as a result the computational time required for carrying out cross validation can be significantly reduced, hence the training time can be much quicker than implementing the sklearn.svm.SVC function. The sklearn.svm.LinearSVC function has been found to be more efficient in comparison when dealing with a large dataset such as document classification.

### SVM Parameters

The parameters that can be tuned while computing a SVM classifier is explained in section 2.1.2. When training the SVM classifier using the sklearn package, the c parameter can be altered by giving a strictly positive number when calling the model training function. The gamma parameter can only be altered for the sklearn.svm.SVC function, for kernels coefficient for 'rbf', 'poly' and 'sigmoid'. Multiple inputs can be used for the gamma parameter:
(i) gamma = 'scale', where the gamma value is set as 1/(n_features*X.var()).
(ii) gamma = 'auto', the gamma value is 1/n_features.
(iii) gamma = float.

## 3.4 KNN in Python

The scikit-learn package is also used to construct the k-NN classifiers used in the paper. The sklearn.neighbors.KNeighborsClassifier function is used, with the 'n_neighbors' parameter set to various integer values to alter the number of nearest neighbours considered for the classifier.

## 3.5 Author Features

In order to train our model based on the chatlogs provided in the dataset, the content of the conversations as well as the properties of the authors need to be converted into vectors. In this paper, two different types of features are captured, the lexical features and behavioural features.

### 3.5.1 Lexical Feature

Lexical features are information extracted from the pure text. Two different types of lexical features are used in the paper, the bag-of-words features and the sentiment features. For the bag-of-words approach, the texts are separated from strings into tokens, which are the minimal meaningful units can be in the form of characters, words, phrases or other forms.

The Google cloud NLP API is used to extract the sentimental lexical features including the sentiment score and the magnitude of the text. A score between 1 and 0.25 indicate positive sentiment, between 0.25 and -0.25 for a neutral sentiment while between -0.25 and -1 provides a negative sentiment. The magnitude shows how strong the mood is (both positive and negative) within the given text, a value from 0 to +infinity can be yield while the larger the magnitude value, the stronger the mood. The magnitude value is accumulated within the text with both positive and negative emotion, as a result longer text block might lead to a greater magnitude.

### 3.5.2 Behavioural Feature

In addition to the lexical features of the text, the behavioural pattern of the authors can be synthesised. In order to differentiate between predator and victim, several different features are collected for the authors.
(i) The percentage of conversation started by the author.
(ii) Average time an author takes to reply to the message.
(iii) The number of conversation an author participates in.

(iv) Number of messages sent by an author.

(v) The percentage of lines sent by an author.

(vi) The number of characters sent by an author.

(vii) The percentage of characters sent by an author.

(viii) The number of unique authors an author chat with.

### 3.5.3 Features Standardisation

The lexical and behavioural features of the obtained would be in different scales with various means, which would result in a biased weighing in the training stage. Thus the features extracted are standardised using the sklearn.preprocessing.StandardScaler function to obtain a mean = 0 and variance = 1.

## 3.6 Dimension Reduction

The training data used in the paper consists of high dimensionality of on the order of over 100,000 lexical features and behaviour features, which would result in an increase in the computational time required as well as the potential co-linearity in the features. (Li, 1991) suggested that it would be useful for the interesting features of the high dimensional data being able to be retrieved from their low dimensional projections. The concept of dimension reduction is introduced, as stated by (Kambhatla & Leen, 1997) the objective of dimension reduction is to obtain a parsimonious description of multivariate data. Where the goal is to reduce the redundancy in the features, to eliminate all but one co-linear feature to obtain an accurate and clear set of components.

Principal component analysis (PCA) is a tool that reduces a complex dataset into a lower dimension. (Shlens, 2014) states that the goal for PCA is to identify the most meaningful features to re-express the dataset. The procedure of carrying out a PCA is given in the paper as:

(i) Organise the data into a $m \times n$ matrix, where $m$ is the number of features and $n$ is the number of instances.

(ii) Standardise the features.

(iii) Calculate the covariance matrix.

(iv) Calculate the principal components of the covariance.

(v) Choosing components for new features.

The percentage of the variance each principal component can be account for indicates the percentage of the data that can be explained by such principal components. The larger the number is, the more significant the relevant feature is for the classifier. Principle components that obtain small numbers are normally removed to reduce the feature size.

#### Dimension Reduction in Python

Dimension reduction can be carried out in python using multiple functions including the sklearn.decomposition.PCA and sklearn.preprocessing.StandardScaler for standardise the features in the training set (mean = 0 and variance = 1). The pca.fit_transform function can be used to fit the model with the features and apply dimension reduction on them. The number of features remaining from the PCA can be selected using the 'n_components' parameter setting.

## 3.7 Evaluation metric

The performance of a classification model can be evaluated using several different parameters. The prediction outcome can be classified as true positive (TP), false positive (FP), true negative (TN) and false negative (FN) as shown in figure 3.1. The accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{3.1}$$

the accuracy of the model is the percentage of correctly predicted entities out of all entities. The precision is given by:

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{3.2}$$

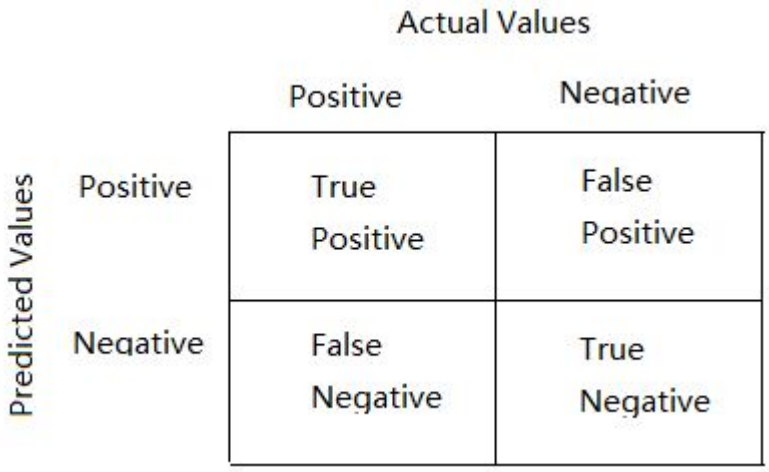


Figure 3.1: The confusion matrix for model performance evaluation

which can be explained as the percentage of correctly predicted positive values out of all the entities that have a positive predicted value. On the other hand, recall is calculated by:

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{3.3}$$

recall is the percentage of the correctly predicted positive values out of all the entities that have a positive actual value.

Based on the nature of the PAN12 dataset, both the training set and the testing set are significantly imbalanced. For the training set 142 out of the 97,689 authors are predators, for the testing set 250 out of 10,000 authors are predators. As a result, the accuracy of the model would not be a representative evaluation of the performance of the model, as by simply labelling all the authors as negative, the model would achieve a 98% accuracy.

The goal of the paper is to correctly identify all the predators in the corpus while reduce the number of falsely predicted non-predators. Hence the precision and recall are combined to give the overall evaluation of the model performance, the F-score is used, which is calculated as suggested by (Schütze, Manning, & Raghavan, 2008):

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1 - \alpha}{\alpha} \tag{3.4}$$

P is the precision and R is the recall, $\alpha \in [0, 1]$ and $\beta^2 \in [0, \infty]$. With the value of $R = 1$, the F value is balanced between the precision and recall, with $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$. By decreasing the $\beta$ value, the precision value in the F-score is emphasised, while by increasing the $\beta$ value the recall score is emphasised.

For the case of the dataset used in the paper, the precision would be the percentage of correctly identified predators out of all the authors that are predicted to be predators. The recall is the percentage of correctly identified predators out of all the actual predators in the corpus. The purpose of developing a classification system for predator is for faster automatic detection of the predator, as well as the reducing of human force required for the conviction process. (Inches & Crestani, 2012) stated that for real-life scenarios, any potential predator identified automatically by the system will be sent to the police agent, and be manually filtered to obtain the actual predators. As the falsely identified predators by the system will result in the demand of human force which can be reduced by increasing the precision of the system. Hence when using the F-score for model evaluation, the precision score is prioritised, with a $\beta$ value of 0.5 selected.

## 3.8 Statistical Significance Test

When evaluating the performance of machine learning models, (Pereira, Mitchell, & Botvinick, 2009) suggests a null hypothesis is normally defined based on the circumstances of the task, and tested using suitable methods for a statistically significant result to accept or reject the null hypotheses.

### 3.8.1 Choosing the Correct Statistical Test

The process of selecting the correct statistical test is given by (McCrum-Gardner, 2008), where the scale of measurements of the data (nominal, ordinal and interval) and the nature of data used for analysis (independent and paired groups) are taken into consideration.

### Scale of Measurements

**(i)Nominal**: data that can be assigned into categories, such as gender (male/ female), country of origin (Ireland, Uk, etc).

**(ii)Ordinal**: data are categorical and associated with the ranked variable, and can be arranged in a certain order such as scale (strongly disagree/ disagree/ agree/ strongly agree)

**(iii)Interval**: data are numerical with meaning associated with the value of the data, hence data values can be compared, such as age, weight etc.

### Comparison between Samples

**(i)Independent groups**: the groups are not related.

**(ii)Paired groups**: the samples in the two groups follow a one to one correspondence, where a sample in the first group can be uniquely paired to a sample in the other group.

| Scale of Measurements | Independent Samples | Paired Samples |
|---|---|---|
| Interval (parametric) | Independent t-test | Paired t-test |
| Ordinal or interval (non-parametric) | Mann-Whitney U-test | Wilcoxon signed rank test |
| Nominal two categories | $\chi^2$-test | McNemarś test |
| Nominal multiple categories | $\chi^1$-test | - |

Table 3.1: Test for comparison between two groups

The way to select the suitable test is given by (McCrum-Gardner, 2008), the method of choosing the correct comparison test between two group is given in table 3.1 and the method of selecting the appropriate test for more than two groups is given in table 3.2. For the case of comparing the performance of two different classifiers on the same dataset, in the paper a 10 fold cross validation is used where the ten f-scores obtained follows a Gaussian distribution. The cross validation score satisfies the condition of parametric measurements obtained from paired samples, as a result the paired t-test can be used to validate if the mean f-score between the two classifiers

are statistical significantly different (Dietterich, 1998). For the case of comparing the performance of classifiers constructed using different training set, the one-way ANOVA test can be used.

| Scale of Measurements | Independent Samples | Paired Samples |
|---|---|---|
| Interval (parametric) | One-way ANOVA | Repeated measures analysis of variance |
| Ordinal or interval (non-parametric) | Kruskal-Wallis one-way ANOVA | Friedmanś test |
| Nominal | $\chi^2$-test for RXC table | Cochranś Q |

Table 3.2: Test for comparison between more than two groups

### 3.8.2 Paired t-test

The numeric formula for paired t-test is given by (Field, Miles, & Field, 2012) as the following:

$$t = \frac{\overline{D} - \mu_{\mathrm{D}}}{s_{\mathrm{D}}/\sqrt{N}} \tag{3.5}$$

$\overline{D}$ is the mean difference between the actual samples, $\mu_{\mathrm{D}}$ is the expected difference between the population means and $s_{\mathrm{D}}/\sqrt{N}$ is the standard error of the differences, where $s_{\mathrm{D}}$ is the standard deviation, $N$ is the sample size. (Menke & Martinez, 2004) states several assumptions for the paired t-test need to be satisfied:

(i) The data are normally distributed

(ii) The data are continuous

(iii) The data are independent

For the case of the paper, the performance of the models are evaluated using the f-score, which satisfies the data are continuous assumption. A k fold cross validation is carried out in the training corpus, as a result, each sample is used in the training set for (k-1) times, thus strictly speaking the data are dependent which violates the third assumption of independence, which would lead to an optimistic result that involves a

higher Type I error (Falsely reject of a true null hypothesis). However, as stated by (Dietterich, 1998) although k fold cross validation can lead to a high type I error, it also results in a relatively low type II error (Failure to reject the false null hypothesis), as a result it is recommended in the cases when type II errors are more important. The normality of the data distribution can be tested, which can be further normalised to reach the assumption of the paired t-test.

The p-value is yield from the paired t-test and can hence be used to evaluate if the means are significantly different. The definition for the p-value is given by (Goodman et al., 1999) as the probability of obtaining a result equal to the observed result under the null hypothesis, or in other word an informal measure of the difference between the null hypothesis and the data. For the case of the paired t-test in the paper, at a 95% confidence interval, a p-value less than 0.05($\alpha$ level) suggests the means are statistical significantly different, and hence rejects the null hypothesis.

### 3.8.3 Paired t-test in Python

#### Normality Test

The normality of the k fold cross validation result can be tested using the python package 'scipy' with a built in function scipy.stats.normaltest [6]. The function tests the null hypothesis that the sample array is normally distributed. By feeding the array of cross validation results into the function, it returns the p-value which is a two sided $\chi^2$ probability for the hypothesis test, with the p-value smaller than the $\alpha$ level (normally = 0.05), the null hypothesis can be rejected, for the case of p greater than the $\alpha$ level, the null hypothesis can not be rejected.

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html#r7bf2e556f491-1

**Paired t-test**

The 'scipy' package also consists of a function for paired t-test: scipy.stats.ttest_rel [7]. By feeding two arrays of input samples, a t-statistic value and a two sided p-value can be yield from the function. The null hypothesis is defined as the means of the two arrays are not statistical significantly different, with a p-value greater than the $\alpha$ value (normally = 0.05) the null hypothesis can not be rejected, otherwise, with a p-value less than the $\alpha$ value, the evidence is statistically enough to reject the null hypothesis.

### 3.8.4 One-way ANOVA

For a one way analysis of variance (ANOVA) model, with $\mu_1, ..., \mu_j$ being the means of j independent variables with standard deviations $\sigma_1, ...., \sigma_j$, the null hypothesis can be given as (Wilcox, 1989):

$$H_0 : \mu_1 = \mu_2 = .... = \mu_j \tag{3.6}$$

where the alternate hypothesis is stated as there consists at least one different $\mu$ value between the groups. The F-test is used to test if there are statistically significant difference between the means.

The one-way ANOVA can be carried out in Python using the statsmodels.formula.api.ols function from the 'statsmodels' model (Seabold & Perktold, 2010). The F-statistic score and the p-value can be obtained in the result. For a p-value smaller than 0.05, the null hypothesis can be rejected, i.e. there are statistically significant difference between the means of the groups. On the other hand, the null hypothesis can not be rejected for p-value greater than 0.05. The statistical significant difference between each group in the ANOVA can also be obtained using the t_test_pairwise function, to show if all the groups involved in the test are statistical significantly different from each other.

[7] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html

## 3.9 Experiment Design

### 3.9.1 Suspicious Conversation Extraction

For the pre-processing stage of the training set, the mis-spellings and typos are not corrected, stemming is also not carried out due to the nature of the dataset. As stated by (Peersman, 2018), the mis-spellings in the chat logs, either intentional or unintentional can be representative of an author's chatting habit, which might also contain potential information about the author's age and gender information that can be extracted using machine learning methods. Stop words and commonly used words (eg: 'hi', 'hey') are also not removed, as such words can contain behavioural feature of the author, such as the number of time an author starts a conversation, the average time an author takes to reply to messages etc. The size of the training set is reduced by re-balancing the number of instances between the majority and minority classes through downsampling, the details will be given in section 3.9.2.

The first stage of the classification process is to identify the suspicious conversation which is defined as the conversation participated by a predator. The training set is relabelled as suspicious and unsuspicious conversations, where 2016 conversations from the training set are labelled as suspicious conversations. The conversations are classified by performing the tf-idf of the n-grams for the SVM classifiers, a 10 fold cross validation is used to find the optimum classifier. The method of finding the best parameters and kernel setting for the SVM classifier is explained in the following section. Where the optimum parameters obtained are used to compare the performance between using a character n-gram and a word n-gram.

### 3.9.2 Unbalanced Dataset and SVM Parameters

The properties of an unbalanced dataset have been explained in section 2.3, as well as the potential affect using an unbalanced training set has on the performance in the model. The training set used in the paper is extremely unbalanced, the following ex-

periment is carried out to test the effect of having different ratio between the number of instances in each class.

The suspicious conversations and unsuspicious conversations in the training set are separated, with conversations randomly selected from the group of unsuspicious conversations and combined with all the suspicious conversations to generate training sets with different ratios between the two types. Training sets with the ratio between suspicious conversations and unsuspicious conversations of (1:1, 1:2, 1:4, 1:8 and 1:16) were generated by combining 2,016 suspicious conversations with randomly selected unsuspicious conversations of number (2,016, 4,032, 8,064, 16,128 and 32,256). The training sets generated are then used in finding the effect of re-balancing the dataset.

Using the training set with the ratio between suspicious and unsuspicious being 1:1, the optimum parameter setting and kernel choice is tested for the SVM classifiers. The performance of the different configurations are tested by performing a 10 fold cross validation on the n-gram tf-idf to construct SVM classifiers for suspicious conversation identification. Models are built using the sklearn.svm.SVC function and the sklearn.svm.LinearSVC function, for the case of the SVC function, all the kernel types are tested ('linear', 'poly', 'rbf' and 'sigmoid'), for the case of 'rbf', 'poly' and 'sigmoid' kernel, the gamma parameter is tested between 'scale' and 'auto'. For each set of configuration used, the 10 fold cross validation is repeated 50 times using various c value from 0.1 to 5 with 0.1 interval. A paired t-test is then carried out between the two models with the best performance, to see if a model with certain configuration setting is statistical significantly better than the other.

With the optimum kernel option, gamma and c parameter value conducted in the previous paragraph, the configuration setting is used to compare the effect of training set with different ratio between the suspicious conversations and the unsuspicious conversations (1:1, 1:2, 1:4, 1:8 and 1:16). The previously constructed training sets are used, by using a 10 fold cross validation, the f-score of different training sets are

obtained. The effect of dimension reduction on the training features is also evaluated by building classifiers using different number of features.

### 3.9.3 Age Feature Extraction

In the paper, the age of the authors in the predators dataset is proposed to be extracted from the messages sent by the authors using machine learning methods. The NPS Chat Corpus is used to train the age prediction model. The objective of the paper is to identify the predators whose victims are adolescents based on the definition of predators, thus to classify the author to a specific age group (eg: 20s, 30s, etc) is not necessary, for the paper age classification is only required between teens and adults. Hence only the teens chat room files and the adults chat room files from the NPS Chat Corpus are used in the paper.

The dataset is recomposed into training and testing set, with 1412 messages from teens and 1411 messages from adults combined into the training set and 706 teens post combined with 706 adults messages as the testing set. Due to the nature of the chat corpus, it contains a large number of uninformative messages in both the adults and teens chats such as 'PART' and 'JOIN' representing when a user joins or leaves the chat room, those messages are removed from the corpus in the pre-processing stage, resulting in 958 teens and 1052 adults messages left for the training corpus and 454 teens and 633 adults messages left for the testing corpus. Similar to the PAN12 corpus, no further pre-processing stages are carried out. The SVM model is trained using the 'LinearSVC' configuration with a 10 fold cross validation. A range of values are tested for the c parameter of the SVM classifier from 0.1 to 5 with 0.1 interval. The c parameter that provides the optimum performance is used to train the final model, which is tested on the testing corpus from the NPS Chat Corpus, and the PAN12 corpus. For the PAN12 corpus, the age of the authors are not provided, as a result, assumptions can only be made from the definition of the predators, where the predators are adults and the victim are teens. Hence a testing set can be constructed from the training corpus of the PAN12 dataset, by labelling the predators as adults,

and authors participated in the conversation with the predators (victims) are labelled as teens.

### 3.9.4 Predator Identification

After building the model for suspicious conversation identification, the classifier is used to identify the suspicious conversations in the test set. The test set is further modified for the second iteration of classification by removing all the conversations that are predicted as non-suspicious, as such conversations are assumed to contain only non-predators.

The second iteration of classification is be carried out to identify the predators in the conversations. Unlike the identification of suspicious conversations, the predator identification uses both the lexical and behavioural features of the authors. Behavioural features as mentioned in section 3.5.2 are extracted for each author, combined with sentimental features extracted using the Google cloud NLP API including:
(i) Percentage of line sent by an author with negative sentiment.
(ii) Percentage of line sent by an author with positive sentiment.
(iii) Percentage of line sent by an author with neutral sentiment.
(iv) The average magnitude of an author.
(v) The average negative score of an author.
(vi) The average positive score of an author.

The features are extracted and saved in a csv file. Two classifiers are trained:(i) Using only the tf-idf of the n-grams and (ii) Using the n-gram tf-idf combined with the standardised lexical and behavioural features extracted from the authors. The performance of the two SVM classifiers are compared to explore if the performance of the classifier would be improved by adding the behavioural features into the training set. K-NN classifiers are trained using different number of k, to compare the performance between SVM classifiers and the k-NN classifiers.

## 3.10 Summary

In this chapter, the two datasets used for the purpose of the research are explained in detail. The classification techniques are also illustrated in the chapter, with the Python packages and functions used to build the models. The choice of the evaluation metric and statistical significance test are also explained. The experiment design is provided in the chapter including the two stage classifier of suspicious conversation and predator identification, the effect of re-balancing an imbalanced dataset, selecting the optimum parameter and kernel for the classifiers as well as the model for age group classification. The results will be illustrated and evaluated in the next chapter.

# Chapter 4

# Results, evaluation and discussion

This chapter discussed the choice of the parameters and kernel choice for the classifiers used in the experiment. The performance of re-balancing the unbalanced dataset by down-sampling is illustrated. Details of the age identification model built and the performance of the model are discussed. The overall performance of the two stage predator identification model is evaluated. The results obtained are compared with findings from the literature review.

## 4.1 Parameter and Kernels

Parameters and kernel options can be modified for SVM classifiers to achieve optimum performance, the detail have been explained in section 3.9.2. The newly constructed training set with a 1:1 ratio between suspicious and unsuspiscious conversations is used in the section. The tf-idf of the word n-gram with n=1,2 is used as the training features. The f-score and the best c values are shown in table 4.1 for different kernel and gamma settings, a full table of f-scores for different c values can be found in the appendix. The performance of different kernel types is evaluated based on the following standards:

(i) Computational cost, the time required to complete the 10 fold cross validation.

(ii) The f-score.

The performance of different SVM kernel options are explained in section 3.3 as when

| **Linear** SVC | **linear** | **rbf** 'scale' | **rbf** 'auto' |
|---|---|---|---|
| c=0.4 f=0.949 | c=0.7 f=0.9503 | c=1 f=0.9603 | c=0.5 f=0.3254 |
| **sigmoid** 'scale' | **sigmoid** 'auto' | **poly** 'scale' | **poly** 'auto' |
| c=0.2 f=0.9397 | c=0.5 f=0.3254 | c=4.4 f=0.9216 | c= 0.5 f=0.3254 |

Table 4.1: F-score with the optimum c value for classifiers using different kernel and gamma settings. With the ratio between suspicious and non-suspicious conversations being 1:1.

the corpus size and feature dimension increases, the computational time required for training a SVM classifier increases significantly, where the sklearn.svm.LinearSVC function does not require the use of a kernel, as a result, the computational time required for carrying out cross validation is significantly shorter. For the case of the larger training set constructed (1:16 between suspicious and unsuspicious conversations), the computational time required for carrying out a 10 fold cross validation for the LinearSVC function is around 10s, where for the sklearn.svm.SVC function with the 'rbf' kernel setting, the computational time required is around 1700s. On the other hand, when considering the f-score achieved for classifiers, according to table 4.1 the 'rbf' kernel with gamma configuration of 'scale' and c=1 obtained the highest f-score of 0.9603.

Based on the previously mentioned standards for choosing the optimum kernel, a paired t-test is carried out between the 10 fold cross validation score of the linearSVC classifier (with significantly lower computational time) and the 'rbf' kernel with gamma setting as 'scale' (highest f-score obtained). A t-statistic of -2.018 and p-value of 0.027 is obtained, which is lower than the $\alpha$ level (0.05), the null hypothesis of not having

| | 1:1 | 1:2 | 1:4 | 1:8 | 1:16 |
|---|---|---|---|---|---|
| Avg F-score | 0.9603 | 0.9384 | 0.9127 | 0.8858 | 0.8483 |

Table 4.2: F-score for imbalanced datasets

statistical significant difference between the mean of the two 10 fold cross validation results can be rejected. In other word, according to the statistic test, the 'rbf' kernel with gamma setting 'scale' has a statistical significantly better performance than the LinearSVC classifier. Hence, for the SVM used in the following section, the svm.SVC function with 'rbf' kernel fuction, c=1 and gamma parameter setting as 'scale' is used.

## 4.2 unbalanced Dataset

The parameter settings and kernel choice from section 4.1 are used in the following sections. The effect of re-balancing a training set is tested, with ratio between suspicious and unsuspicious conversations being 1:1, 1:2, 1:4, 1:8 and 1:16. The average f-score for the 10 fold cross validation using the training sets are shown in table 4.2 (a complete list of the f-score can be found in A.2 using the 'rbf' kernel, with c=1 and gamma parameter set as 'scale'. A one-way ANOVA test is carried out between the cross validation f-scores obtained for the training sets. An f-statistic score of 58.75 and p-value of 2.72e-17 is obtained, the null hypothesis of no statistically significant difference between the performance of the classifiers can be rejected. It can be seen from table 4.2 that the greater the ratio between the classes are, the more unbalanced the dataset becomes, the smaller the average f-score obtained becomes.

A paired test for statistically significant difference between the classifiers trained on training set of different imbalanced ratio is shown in table 4.3. The result shows that all the null hypothesis can be rejected, i.e. the f-scores obtained from classifiers trained using the training sets with different ratio between suspicious and unsuspicious conversations are all statistical significantly different. According to table 4.2 the f-score increases each time when the difference between the two classes decreases, as a result

| | p-value | reject $H_0$ |
|---|---|---|
| 1:16-1:1 | 0.0e+00 | True |
| 1:2-1:1 | 9.716e-03 | True |
| 1:4-1:1 | 2.45e-06 | True |
| 1:8-1:1 | 5.5e-11 | True |
| 1:2-1:16 | 1.58e-13 | True |
| 1:4-1:16 | 2.98e-09 | True |
| 1:8-1:16 | 1.3e-04 | True |
| 1:4-1:2 | 5.52e-03 | True |
| 1:8-1:2 | 3.52e-07 | True |
| 1:8-1:4 | 5.35e-03 | True |

Table 4.3: A paired test for statistically significant difference between the classifiers trained on training set with different ratio between suspicious and unsuspicious conversations.

by down sampling the majority class of the training set, the model would conduct a better classification performance on the desired minority class. As the classifier trained with the balanced dataset (with 1:1 ratio between the suspicious and unsuspicious conversations) performed statistical significantly better than the rest, it will be used as the training set for the rest of the paper.

Dimension reduction is also performed on the dataset, with the number of components left being 250, 500 and 1000. The f-scores of the 10 fold cross validation can be find in table 4.4. As the result shows, reducing the dimensionality of the features used to train the model decreases the performance of the model, which agrees with the findings from Tello. Based on the observations, dimension reduction is not used in for the classification algorithms.

| Number of components | 250 | 500 | 1000 | all |
|---|---|---|---|---|
| F-score | 0.953 | 0.954 | 0.956 | 0.96 |

Table 4.4: F-score for classifier after dimension reduction using different number of components

| | Word | Character |
|---|---|---|
| Uni-gram | 0.9531 | 0.8944 |
| Bi-gram | 0.9365 | 0.949 |
| Tri-gram | 0.8581 | 0.961 |
| n=(1,2) | 0.9603 | 0.9434 |
| n=(1,3) | 0.9583 | 0.9568 |

Table 4.5: F-score for word and character n-grams with different n values

## 4.3 N-Gram

The performance of the classifier using the tf-idf of different n-gram is tested with word and character unigram, bigram, trigram with the combination of unigram and bigram as well as all the three techniques. The best f-score is obtained for using the character trigram, with the second best f-score obtained using a combination of word unigram and bigram. A paired t-test is carried out between the two classifiers, giving a result of a t-statistic of -0.167 and p-value of 0.871. The p-value is larger than the $\alpha$ level (0.05) indicating there is no statistically significant difference between the classifiers. Hence, the combination of word unigram and bigram is used for the following sections.

## 4.4 Age Identification

The age identification classifier is built using data from the NPS Chat Corpus to predict the age group (teens or adults) of the author. A 10 fold cross validation is carried out on the training set generated from the NPS Chat Corpus, the c parameter for the SVM classifier is tested using a range of value from 0.1 to 5 with 0.1 interval to find

| | Actual Teens | Actual Adults |
|---|---|---|
| Predicted Teens | 241 | 162 |
| Predicted Adults | 392 | 471 |

Table 4.6: Age identification result for NPS Chat Corpus

the optimum model. The accuracy score is used to evaluate the performance of the model, as the main task is to correctly classify the age of the author. For the cross validation, the highest accuracy is achieved as 0.763 with a C parameter of 1.8.

The model is further tested on the testing set constructed from the NPS Chat Corpus and the testing set generated from the PAN12 predators dataset, the results are shown in table 4.6 and table 4.7. The model obtained a 65% accuracy on the testing set generated from the NPS Chat Corpus, it can be seen from table 4.6, the performance of the model is not ideal, however, the model is not significantly biased, as 74.4% of the adults and 38% of the teens are categorised correctly. For the case of the PAN12 testing set, the model only achieved a 51.4% accuracy, however, it can be seen from table 4.7 that 140 out of 142 of the predators are correctly classified as adults, while only 4 out of the 138 victims are classified as teens.

Based on the nature of the PAN12 predator dataset, the pseudo victims in the predator chat logs are volunteers posing to be adolescents, who are actually adults in real life. Although the pseudo victims' real age profile were not successfully identified by the predators (the volunteers are treated as adolescents by the predators), 97.1% of the pseudo victims are classified as adults by the classifier which agrees with their real age identity.

Due to the low accuracy of the age identifier, it can not be applied in the paper for the purpose of predicting the age of the authors in the PAN12 corpus. However, the age identifier does show great potential for future work on areas such as false online profile detection, despite the fact humans might not be able to identify an author's age

| | Actual Teens | Actual Adults |
|---|---|---|
| Predicted Teens | 4 | 2 |
| Predicted Adults | 134 | 140 |

Table 4.7: Age identification result for PAN12 Corpus

based on posts or messages, it is possible that with the aid of computational power, the author's age can be identified using the lexical features of the messages sent by the person.

## 4.5 Suspicious Conversation Identification

A model is built using the training set with 1:1 ratio between the suspicious and unsuspicious conversations, using the 'rbf' kernel with c=1 and gamma parameter set to 'scale'. The tf-idf of the combination of word unigram and bigram is used to train the model for suspicious conversation identification. The model is then used to predict the potential suspicious conversations in the test set, the confusion matrix of the classification results is shown in table 4.8. The classifier obtained an accuracy of 0.947 and f-score of 0.349.

The testing set consists of 3,737 suspicious conversations and 151,391 unsuspicious conversation, where 10,790 conversations are predicted as suspicious and 144,338 conversations are predicted as unsuspicious. Although the f-score obtained is not ideal, 3,350 out of 3,737 suspicious conversations are successfully identified. The testing set consists of 254 predators in total, where the 3,350 successfully identified suspicious conversations consists of 251 predators and 10,583 non-predators. The test set is further modified by removing all the conversations that are predicted as non-suspicious.

| | Suspicious | Unsuspicious |
|---|---|---|
| Predicted Suspicious | 3,350 | 7,440 |
| Predicted Unsuspicious | 387 | 143,951 |

Table 4.8: Confusion matrix for suspicious conversation identification

| Feature type | tf-idf | behavioural | lexical | all |
|---|---|---|---|---|
| f-score | 0.81 | 0.57 | 0.109 | 0.598 |

Table 4.9: F-score obtained by classifiers trained with only n-gram tf-idf, only behavioural features, only lexical features and all together

## 4.6 Predator Identification

The lexical and behavioural features of the remaining conversations are extracted using the method illustrated in section 3.9.4 and saved in a csv file. Classifiers are trained using only the tf-idf of n-grams, only the lexical features, only the behavioural features and the last one using the n-grams tf-idf combined with the lexical and behavioural features extracted for each author. The f-scores of the classifiers are shown in table 4.9. The classifier trained with only the n-gram tf-idf achieved an accuracy of 0.989 with f-score of 0.81, while the classifiers trained with only behavioural and lexical features obtained f-scores of 0.57 and 0.109 the classifier trained with additional lexical and behavioural features achieved an accuracy of 0.9816 and f-score of only 0.598. The result indicates that the n-gram tf-idf contributes the most to the performance of the classifier,

| | Predator | victims | other |
|---|---|---|---|
| Predicted Predator | 171 | 20 | 10 |
| Predicted Non-predator | 80 | 155 | 10,398 |

Table 4.10: Confusion matrix for predator identification using only n-gram tf-idf

| | Predator | Non-predator | other |
|---|---|---|---|
| Predicted Predator | 121 | 62 | 7 |
| Predicted Non-predator | 130 | 133 | 10,381 |

Table 4.11: Confusion matrix for predator identification using n-gram tf-idf combined with author lexical and behavioural features

| k | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| f-score | 0.64 | 0.679 | 0.706 | 0.703 |

Table 4.12: F-scores obtained for k-NN classifiers with different k parameter

The confusion matrices of the two classifiers using only n-gram tf-idf and the second one combined n-gram tf-idf with lexical and behavioural features are shown in table 4.10 and table 4.11. Comparing the performance of the two classifiers, the classifier used only the n-gram tf-idf successfully identified 171 out of the total of 251 predators in the modified testing set with 80 false negatives and 30 false positives, 20 of the false positives are victims. On the other hand, the classifier trained using additional lexical and behavioural features of the authors identified 121 predators with 130 false negatives and 69 false positives with 62 of the false positives being victims.

K-NN classifiers are trained using only n-gram tf-idf as input with different number of nearest neighbours, the f-scores obtained are shown in table 4.12. The classifier with k=4 obtained the best performance with an f-score of 0.706, with 92 out of 251 predators successfully identified. Hence the SVM classifier performed better than the k-NN classifier.

When considering the overall performance of the two stages classification, 171 out of the total of 254 predators are successfully identified, with 30 false positives and 83 false negatives, resulting in a precision of 0.85, recall of 0.67 and an f-score of 0.807 with $\beta = 0.5$.

## 4.7 Discussion

For the first stage of the classification, a suspicious conversation identification classifier is built and used to find the optimum parameter and kernel setting of the model. The optimum performance is achieved using a 'rbf' kernel with c=1 and gamma='scale'. The performance of re-balancing the dataset through downsampling is test, with the best performance achieved by the classifier trained using the most balanced training set.

An age group classifier is built to categorise the authors as teens and adolescents, the model performed poorly when testing with the PAN12 predator identification dataset, with an accuracy of 51.4%. However, when evaluating the confusion matrix of the model, it successfully predicted 140 out of 142 predators as adults, on the other hand, 134 out of the 138 pseudo-victims (adults posing to be teens) are classified as adults. It agrees with the findings from (Peersman, 2018), when the author participates in a conversation while posing as a different age group, although human participants might not be able to identify the real identity of the author, the author's real age group can be identified using machine learning techniques. The finding can be implemented in the future for false user profile identification.

For the second stage of the classification, the predator identification stage, a f-score of 0.81 is achieved using the SVM classifier which is statistical significantly better than the k-NN classifier with the f-score of 0.706.

The overall classification model achieved a f-score of 0.807 with $\beta$=0.5, comparing to the results achieved by the participants in the PAN12 competition, the result would end up the seventh in the competition. Tello's work ranked first in the competition with a f-score of 0.9346, comparing Tello's work with the experiment carried out in the paper, apart from the different choice on the classification algorithms used, Tello's approach did not address the unbalanced dataset in any ways. A possible explanation

for the lower f-score obtained in the paper is from the feature lost when the training set is downsampled. An alternative upsampling approach could be used in future works.

# Chapter 5

# Conclusion

## 5.1 Research Overview

In the paper a two stage classification model is built to identify online sexual predators by initially finding suspicious conversations and hence, find the predators in the suspicious conversations. Due to the significant difference between the number of instances between the two classes, the performance of re-balancing the training set is tested by downsampling the majority class. An age group classification model is also built to extract the age profile of the authors for potential use in the predator identificaiton model.

The final two stage classification model is trained using both the tf-idf of the n-grams and the extracted behavioural features of the authors as well as the n-grams and behavioural features in isolation. The performance of the classifiers are compared, and compared with the results achieved by the participants in PAN12 competition.

## 5.2 Problem Definition

The dissertation looked at the solution for online sexual predator identification from online chat logs. The objective of the paper is to build a SVM classifier for predator identification, to test the performance of re-balancing an unbalanced dataset through

downsampling and to test whether the age profile of the participants in the chat logs can be extracted and be used to train the predator identification classifier. An age group classification model is built and tested on the PAN12 predator corpus. Previous literature on the subject is reviewed, a two stage classifier is built for the predator identification task and compared with previous results.

## 5.3 Design/Experimentation, Evaluation & Results

The SVM classification algorithm is used for the dissertation, with the optimum parameter settings and kernel options tested to achieve the best performance. Training sets with different ratio between the suspicious and non-suspicious conversations are constructed from the original training set, classifiers are trained and the performances are compared. Results indicating that a statistical significantly better performance is obtained by classifiers trained using the more balanced datasets.

An age group classification model is built using the NPS Chat Corpus, and tested on the PAN12 training set. Although the model obtained a poor performance with a classification accuracy of 51.4%, 140 out of 142 predators are successfully classified as adults, while 134 out of 138 pseudo-victims (adults posing as teens) are also categorised as adults. The results agrees with the finding from (Peersman, 2018) that machine learning algorithms may be able to identify the authors using a false profile, i.e. posing to be from a different age group.

Dimension reduction is also carried out to reduce the computational cost of the algorithm by decreasing the number of features used to train the model. A decrease in the performance of the classifier is obtained after the dimension reduction process, which agree to the finding from (Villatoro-Tello et al., 2012) that decreasing the number of features used to train the classifier would also omit potential useful features.

Classifiers are trained for the predator identification stage using only the tf-idf of

n-grams, the extracted author features and everything combined. Similar to the finding from (Morris, 2013; Van Hee et al., 2015), the tf-idf n-gram contributes greatly to the performance of the classifier, where the features extracted from the authors has limited contribution.

The final two stage classifier successfully identified 171 out of 254 predators, resulting in a precision of 0.85, recall of 0.67 and f-score of 0.807 with $\beta$=0.5.

## 5.4 Contributions and impact

The results in the paper contributes to the conclusion that online sexual predators can be identified from online chat logs by performing text classification. This study shows that the wording habit of the author (n-gram) contributes more to the identification of the predator instead of the behavioural features (time taken to reply to message, number of times an author starts a conversation etc). Additionally the re-balancing of the unbalanced dataset in the study by downsampling shows that classifiers trained on balanced dataset obtains a statistical significantly better performance, however, the downsampling of the sample could result in a lost in content, especially when the minority sample size is significantly small.

For the age identification experiment, although the result obtained from the paper is not ideal, it is interesting to see that machine learning algorithms might be able to identify an author's real age group, which can not be easily identified by human.

## 5.5 Future Work & recommendations

Due to ethical reasons, the amount of public predatory conversation is extremely limited. With PAN12 sexual predator identification dataset being the only benchmark dataset with English conversations, the scope of the study is limited to conversations between predators and pseudo-victims between the year of 2004 and 2012. With the

vast changing of internet vocabularies, and phrases used on social medias, the models built from the PAN12 corpus might not be able to function effectively for online sexual predator identification from more recent chat logs. A classifier that can be used for real life predator identification might require more recent data as well as conversations between predators and real victims (under-aged).

Although the age group classification model does not contribute to the predator identification task of the dissertation, it is possible for the model to be used for false profile identification. By taking the messages or posts from the author, the age of the author may be identified and compared with the personal information provided on the profile page.

# References


Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (pp. 39–50).

Arumugam, R., & Shanmugamani, R. (2018). *Hands-on natural language processing with python: A practical guide to applying deep learning architectures to your nlp applications*. Packt Publishing Ltd.

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, *5*(Sep), 1089–1105.

Ben-Hur, A., & Weston, J. (2010). A user's guide to support vector machines. In *Data mining techniques for the life sciences* (pp. 223–239). Springer.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Cavnar, W. B., Trenkle, J. M., et al. (1994). N-gram-based text categorization. In *Proceedings of sdair-94, 3rd annual symposium on document analysis and information retrieval* (Vol. 161175).

Chang, C.-C. (2011). " libsvm: a library for support vector machines," acm transactions on intelligent systems and technology, 2: 27: 1–27: 27, 2011. *http://www. csie. ntu. edu. tw/˜ cjlin/libsvm*, *2*.

Chau, M., & Chen, H. (2008). A machine learning approach to web page filtering using content and structure analysis. *Decision Support Systems*, *44*(2), 482–494.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, *6*(1), 1–6.

Cheong, Y.-G., Jensen, A. K., Gunadóttir, E. R., Bae, B.-C., & Togelius, J. (2015). Detecting predatory behavior in game chats. *IEEE Transactions on Computational Intelligence and AI in Games*, *7*(3), 220–232.

Desmet, B., & Hoste, V. (2014). Recognising suicidal messages in dutch social media. In *9th international conference on language resources and evaluation (lrec)* (pp. 830–835).

Diamantidis, N., Karlis, D., & Giakoumakis, E. A. (2000). Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*, *116*(1-2), 1–16.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, *10*(7), 1895–1923.

Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the detection of textual cyberbullying. In *fifth international aaai conference on weblogs and social media.*

Esposito, L. C. (1998). Regulating the internet: The new battle against child pornography. *Case W. Res. J. Int'l L.*, *30*, 541.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, *9*(Aug), 1871–1874.

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using r*. Sage publications.

Forsythand, E. N., & Martell, C. H. (2007). Lexical and discourse analysis of online chat dialog. In *International conference on semantic computing (icsc 2007)* (pp. 19–26).

Goodman, S. N., et al. (1999). Toward evidence-based medical statistics. 1: The p value fallacy. *Annals of internal medicine*, *130*, 995–1004.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878–887).

Hartmann, J., Huppertz, J., Schamp, C., & Heitmann, M. (2019). Comparing automated text classification methods. *International Journal of Research in Marketing*, *36*(1), 20–38.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hidalgo, J. M. G., & Díaz, A. A. C. (2012). Combining predation heuristics and chat-like features in sexual predator identification. In *Clef (online working notes/labs/workshop).*

Hussain, M., Wajid, S. K., Elzaart, A., & Berbar, M. (2011). A comparison of svm kernel functions for breast cancer detection. In *2011 eighth international conference computer graphics, imaging and visualization* (pp. 145–150).

Inches, G., & Crestani, F. (2012). Overview of the international sexual predator identification competition at pan-2012. In *Clef (online working notes/labs/workshop)* (Vol. 30).

Jeney, P. (2015). Combatting child sexual abuse onling.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Icml* (Vol. 99, pp. 200–209).

Kambhatla, N., & Leen, T. K. (1997). Dimension reduction by local principal component analysis. *Neural computation*, *9*(7), 1493–1516.

Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, *1*(1), 4–20.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).

Kontostathis, A. (2009). Chatcoder: Toward the tracking and categorization of internet predators. In *Proc. text mining workshop 2009 held in conjunction with the ninth siam international conference on data mining (sdm 2009). sparks, nv. may 2009.*

Lameski, P., Zdravevski, E., Mingov, R., & Kulakov, A. (2015). Svm parameter tuning with grid search and its impact on reduction of model over-fitting. In *Rough sets, fuzzy sets, data mining, and granular computing* (pp. 464–474). Springer.

Leslie, C., Eskin, E., & Noble, W. S. (2001). The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002* (pp. 564–575). World Scientific.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, *86*(414), 316–327.

Livingstone, S., Haddon, L., Görzig, A., & Ólafsson, K. (2011). Risks and safety on the internet: the perspective of european children: full findings and policy implications from the eu kids online survey of 9-16 year olds and their parents in 25 countries.

McCrum-Gardner, E. (2008). Which is the correct statistical test to use? *British Journal of Oral and Maxillofacial Surgery*, *46*(1), 38–41.

Menke, J., & Martinez, T. R. (2004). Using permutations instead of student's t distribution for p-values in paired-difference algorithm comparisons. In *2004 ieee*

*international joint conference on neural networks (ieee cat. no. 04ch37541)* (Vol. 2, pp. 1331–1335).

Morris, C. (2013). Identifying online sexual predators by svm classification with lexical and behavioral features. *Master of Science Thesis, University Of Toronto, Canada*.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565.

Onan, A. (2018). An ensemble scheme based on language function analysis and feature engineering for text genre classification. *Journal of Information Science*, *44*(1), 28–47.

Parapar, J., Losada, D. E., & Barreiro, A. (2012). A learning-based approach for the identification of sexual predators in chat logs. In *Clef (online working notes/labs/workshop)* (Vol. 1178).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peersman, C. (2018). *Detecting deceptive behaviour in the wild: text mining for online child protection in the presence of noisy and adversarial social media communications* (Unpublished doctoral dissertation). Lancaster University.

Pendar, N. (2007). Toward spotting the pedophile telling victim from predator in text chats. In *International conference on semantic computing (icsc 2007)* (pp. 235–241).

Pentel, A. (2015). Automatic age detection using text readability features. In *Edm (workshops)*.

Pereira, F., Mitchell, T., & Botvinick, M. (2009). Machine learning classifiers and fmri: a tutorial overview. *Neuroimage*, *45*(1), S199–S209.

Potha, N., Maragoudakis, M., & Lyras, D. (2016). A biology-inspired, data mining framework for extracting patterns in sexual cyberbullying data. *Knowledge-Based Systems*, *96*, 134–155.

Provost, F. (2000). Machine learning from imbalanced data sets 101. In *Proceedings of the aaai'2000 workshop on imbalanced data sets* (Vol. 68, pp. 1–3).

Rangel Pardo, F. M., Celli, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015). Overview of the 3rd author profiling task at pan 2015. In *Clef 2015 evaluation labs and workshop working notes papers* (pp. 1–8).

Schütze, H., Manning, C. D., & Raghavan, P. (2008). Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference* (p. 260).

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th python in science conference.*

Shah, F. P., & Patel, V. (2016). A review on feature selection and feature extraction for text classification. In *2016 international conference on wireless communications, signal processing and networking (wispnet)* (pp. 2264–2268).

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Social, W. A. (2018). Global digital report 2018. *Erişim: https://wearesocial.com/blog/2018/01/global-digital-report-2018*.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111–133.

Sun, A., Lim, E.-P., & Liu, Y. (2009). On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, *48*(1), 191–201.

Tam, J., & Martell, C. H. (2009). Age detection in chat. In *2009 ieee international conference on semantic computing* (pp. 33–39).

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, *28*(4), 667–671.

Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., ... Hoste, V. (2015). Detection and fine-grained classification of cyberbullying events. In *Proceedings of the international conference recent advances in natural language processing* (pp. 672–680).

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Villatoro-Tello, E., Juárez-González, A., Escalante, H. J., Montes-y Gómez, M., & Pineda, L. V. (2012). A two-step approach for effective detection of misbehaving users in chats. In *Clef (online working notes/labs/workshop)* (Vol. 1178).

Wang, H., & Hu, D. (2005). Comparison of svm and ls-svm for regression. In *2005 international conference on neural networks and brain* (Vol. 1, pp. 279–283).

Wilcox, R. R. (1989). Adjusting for unequal variances when comparing means in one-way and two-way fixed effects anova models. *Journal of Educational Statistics*, *14*(3), 269–278.

Yang, Y., Liu, X., et al. (1999). A re-examination of text categorization methods. In *Sigir* (Vol. 99, p. 99).

Yu, H., Ho, C., Juan, Y., & Lin, C. (2013). Libshorttext: A library for short-text classification and analysis. *Rapport interne, Department of Computer Science, National Taiwan University. Software available at http://www. csie. ntu. edu. tw/cjlin/libshorttext*.

# Appendix A

# Additional content

## A.1 F-Score for Different C Parameter

F-scores of the SVM classifiers for suspicious conversations detection using different kernels and gamma settings. The c parameter is altered in the range of 0.1 to 5.0 with 0.1 interval to find the c value for the optimum performance. The f-scores for word n-gram, with n value in the range of 1 to 3 and c value from 0.1 to 2 are shown in table A.1.

| **C** | **Linear** SVC | **linear** | **rbf** 'scale' | **rbf** 'auto' | **sigmoid** 'scale' | **sigmoid** 'auto' | **poly** 'scale' | **poly** 'auto' |
|---|---|---|---|---|---|---|---|---|
| 0.1 | 0.9465 | 0.9393 | 0.893 | 0.3254 | 0.939 | 0.3254 | 0.4433 | 0.3254 |
| 0.2 | 0.9468 | 0.9433 | 0.928 | 0.3254 | 0.9397 | 0.3254 | 0.5793 | 0.3254 |
| 0.3 | 0.9478 | 0.9456 | 0.9411 | 0.3254 | 0.9385 | 0.3254 | 0.6779 | 0.3254 |
| 0.4 | 0.949 | 0.9467 | 0.947 | 0.3254 | 0.9357 | 0.3254 | 0.7094 | 0.3254 |
| 0.5 | 0.9465 | 0.948 | 0.9518 | 0.3254 | 0.9341 | 0.3254 | 0.7287 | 0.3254 |
| 0.6 | 0.9453 | 0.9493 | 0.9541 | 0.3254 | 0.9281 | 0.3254 | 0.7482 | 0.3254 |
| 0.7 | 0.9448 | 0.9503 | 0.9566 | 0.3254 | 0.9208 | 0.3254 | 0.7749 | 0.3254 |
| 0.8 | 0.9429 | 0.95 | 0.9579 | 0.3254 | 0.9141 | 0.3254 | 0.8121 | 0.3254 |
| 0.9 | 0.9429 | 0.9501 | 0.9595 | 0.3254 | 0.905 | 0.3254 | 0.8448 | 0.3254 |
| 1 | 0.9417 | 0.9498 | 0.9603 | 0.3254 | 0.8992 | 0.3254 | 0.8734 | 0.3254 |
| 1.1 | 0.9413 | 0.9488 | 0.9601 | 0.3254 | 0.8936 | 0.3254 | 0.8782 | 0.3254 |
| 1.2 | 0.9401 | 0.9492 | 0.9592 | 0.3254 | 0.8894 | 0.3254 | 0.8799 | 0.3254 |
| 1.3 | 0.9366 | 0.9487 | 0.9591 | 0.3254 | 0.8863 | 0.3254 | 0.8815 | 0.3254 |
| 1.4 | 0.9318 | 0.9479 | 0.9593 | 0.3254 | 0.8837 | 0.3254 | 0.8822 | 0.3254 |
| 1.5 | 0.9294 | 0.948 | 0.9595 | 0.3254 | 0.88 | 0.3254 | 0.884 | 0.3254 |
| 1.6 | 0.9275 | 0.9476 | 0.9596 | 0.3254 | 0.8781 | 0.3254 | 0.8849 | 0.3254 |
| 1.7 | 0.9253 | 0.9471 | 0.9601 | 0.3254 | 0.8757 | 0.3254 | 0.887 | 0.3254 |
| 1.8 | 0.9221 | 0.9464 | 0.96 | 0.3254 | 0.8737 | 0.3254 | 0.8884 | 0.3254 |
| 1.9 | 0.9191 | 0.946 | 0.9596 | 0.3254 | 0.8711 | 0.3254 | 0.8902 | 0.3254 |
| 2.0 | 0.9188 | 0.945 | 0.9596 | 0.3254 | 0.868 | 0.3254 | 0.8925 | 0.3254 |

Table A.1: f-scores for different c parameters for various kernel settings

| | | | | | |
|---|---|---|---|---|---|
| LinearSVC | 0.9729 | 0.9466 | 0.9282 | 0.9442 | 0.9261 |
| C=0.4 | 0.9593 | 0.9432 | 0.9619 | 0.9484 | 0.95933264 |
| rbf | 0.9735 | 0.9443 | 0.9557 | 0.9592 | 0.956429 |
| C=1 | 0.9797 | 0.9575 | 0.9548 | 0.9672 | 0.9546925 |

Table A.2: Cross validation scores for LinearSVC and 'rbf' kernel for training set with 1:1 ratio between suspicious and unsuspicious conversations

| | | | | | |
|---|---|---|---|---|---|
| 1:1 | 0.9735 | 0.9443 | 0.9557 | 0.9592 | 0.956429 |
| | 0.9797 | 0.9575 | 0.9548 | 0.9672 | 0.9546925 |
| 1:2 | 0.9243 | 0.9528 | 0.9398 | 0.9354 | 0.9364 |
| | 0.9352 | 0.9398 | 0.9189 | 0.9384 | 0.9628 |
| 1:4 | 0.9017 | 0.92 | 0.9409 | 0.8983 | 0.9039 |
| | 0.9184 | 0.9179 | 0.9206 | 0.8989 | 0.9063 |
| 1:8 | 0.8522 | 0.8494 | 0.8813 | 0.9041 | 0.9046 |
| | 0.8967 | 0.8921 | 0.8836 | 0.888 | 0.9056 |
| 1:16 | 0.8430 | 0.8947 | 0.8616 | 0.8545 | 0.8277 |
| | 0.8176 | 0.7994 | 0.8578 | 0.8493 | 0.8775 |

Table A.3: Cross validation scores for training set with different ratio between suspicious and unsuspicious conversations

## A.2 Cross Validation Results

The f-scores for the 10 fold cross validation for LinearSVC and 'rbf' kernel for the training set with 1:1 ratio between the suspicious and unsuspicious conversation in table A.2, and the f-scores for the 10 fold cross validation for different ration between the suspicious and unsuspicious conversation in table A.3.